LIGHTWEIGHT PUBLIC KEY INFRASTRUCTURE EMPLOYING UNSIGNED CERTIFICATES

Cross Reference To Related Applications

Sub a1

~~This patent application is related to the following Non-Provisional U.S.~~ Patent Applications: Serial Number XX/XXX,XXX, entitled "AUTHORIZATION INFRASTRUCTURE BASED ON PUBLIC KEY CRYPTOGRAPHY," having Attorney Docket No. 10991054-1; Serial Number XX/XXX,XXX, entitled "METHOD AND APPARATUS FOR PROVIDING FIELD CONFIDENTIALITY IN DIGITAL CERTIFICATES," having Attorney Docket No. 10991055-1; and Serial Number XX/XXX,XXX, entitled "LIGHTWEIGHT PUBLIC KEY INFRASTRUCTURE EMPLOYING DISPOSABLE CERTIFICATES," having Attorney Docket No. 10001540-1, and the following Provisional U.S. Patent Application Serial Number XX/XXX,XXX, entitled "PUBLIC KEY VALIDATION SERVICE," having Attorney Docket Number 10001558-1, which are all filed on even date herewith, are all assigned to the same assignee as the present application, and are all herein ~~incorporated by reference.~~

The Field of the Invention

The present invention relates to public key cryptosystems, and more particularly, to a public key infrastructure employing unsigned certificates for authentication.

Background of the Invention

Public key cryptosystems are globally deployed on the World Wide Web, as well as on a growing number of enterprise networks, for establishment of secure communication channels. Every user in a public key cryptosystem has a pair of keys including a public key and a private key. The public key is disclosed to other users while the private key is kept secret. A public key

cryptosystem typically has a primary designed use, such as for encryption, digital signature, or key agreement. Public key cryptosystems are also used for user authentication. For example, a user can authenticate itself to other users by demonstrating knowledge of its private key, which other users can verify using the corresponding public key.

Sub a2

~~In an application of a public key cryptosystem for authenticating a user,~~ the public key must be securely associated with the identity of the user that owns the public key by authenticating the public key itself. Public key certificates are typically employed to authenticate the public key. A public key certificate is a digital document, signed by a certificate authority, that binds a public key with one or more attributes that uniquely identify the owner of the public key. The public key certificate can be verified using the certificate authority's public key, which is assumed to be well known or is recursively certified by a higher authority. For example, in a corporation, a public key certificate can bind a ~~public key to an employee number.~~

A public key infrastructure (PKI) refers to the collection of entities, data structures, and procedures used to authenticate public keys. A traditional PKI comprises a certificate authority, public key certificates, and procedures for managing and using the public key certificates.

One type of a user of a PKI owns the public key contained in a public key certificate and uses the certificate to demonstrate the user's identity. This type of user is referred to as the subject of the certificate or more generally as the subject. Another type of user relies on a public key certificate presented by another user to verify that the other user is the subject of the certificate and that the attributes contained in the certificate apply to the other user. This type of user that relies on the certificate is referred to as a verifier or relying party.

The association between a public key and an identity can become invalid because the attributes that define the identity no longer apply to the owner of the public key, or because the private key that corresponds to the public key has been compromised. A PKI typically employs two complementary techniques for dissociating a public key from an identity. In the first technique, each public key

certificate has a validity period defined by an expiration date, which is a substantial period from the issue date, such as one year from the issue date. In the second technique, the certificate authority revokes a public key certificate if the public key certificate's binding becomes invalid before the expiration date. One way of revoking a public key certificate is by including a serial number of the public key certificate in a certificate revocation list (CRL), which is signed and issued by the certificate authority at known periodic intervals, such as every few hours or once a day. An entity that relies on a certificate is responsible for obtaining the latest version of the CRL and verifying that the serial number of the public key certificate is not on the list.

CRLs typically become quite long very quickly. When the CRLs become long, performance is severely impacted. First, CRL retrieval consumes large amounts of network bandwidth. Second, each application has to retrieve the CRL periodically, parse the CRL, and allocate storage for the CRL. Then, the application needs to carry out a linear search of the CRL for the public key certificate serial number when the application verifies each public key certificate. As a result, conventional PKIs do not scale beyond a few thousand users.

One solution proposed to alleviate the linear search problem is to partition CRLs. The serial number of the public key certificate determines where the CRL partition is located when the public key certificate is revoked. With partitioned CRLs, the application still has to retrieve and store the entire CRL or else the application needs to fetch a CRL partition in order to verify a certificate. Since certificate verification is a likely critical path, fetching a CRL partition impacts the time it takes to run the application.

An on-line certificate status protocol (OCSP) operates by permitting the verifier of the public key certificate to ask the certificate authority if the certificate is currently valid. The certificate authority responds with a signed statement. The OCSP allows CRLs to be avoided, but requires the verifier to query the certificate authority as part of the transaction that employs the public key certificates. The verifier querying the certificate authority increases the time

it takes to perform the transaction. The OCSP scheme is highly vulnerable to a denial-of-service attack, where the attacker floods the certificate authority with queries. Responding to each query is computationally expensive, because each response requires a digital signature.

In a certificate status proofs scheme, the certificate authority maintains a data structure describing the set of valid and invalid certificates in the directory. For every public key certificate that has not yet expired, a short cryptographic proof can be extracted from the data structure of the certificate's current status (i.e., valid or invalid). A CRL can essentially be viewed as a cryptographic proof of invalidity for the public key certificates in the CRL, and proof of validity for those not in the CRL. The CRL, however, is not a short proof. The short cryptographic proof can be obtained by the verifier from the directory, or it can be obtained by the subject and presented to the verifier together with the public key certificate.

The Simple Public Key Infrastructure (SPKI) working group of the Internet Society and the Internet Engineering Task Force has proposed the possibility of employing short-lived certificates as a method of achieving fine-grain control over the validity interval of a certificate. See C.M. Ellison, B. Frantz, B. Lampson, R. Rivest, B.M. Thomas and T. Ylonen, *SPKI Certificate Theory*, Request for Comments 2560 of the Internet Engineering Task Force, September 1999. The *SPKI certificate theory* reference states that there are cases in which a short-lived certificate requires fewer signatures and less network traffic than various on-line test options. The use of a short-lived certificate always requires fewer signature verifications than the use of a certificate plus on-line test result.

Nevertheless, no practical method of issuing short-lived certificates has been proposed. Traditional certificates are issued off-line, as part of a process that includes subject registration, at the rate of one per year per user. By contrast, short-lived certificates would have to be issued on-line at the rate of at least one per day per user, and perhaps as often as one every few minutes for every user.

The term on-line and the term off-line have particular definitions in the context of a PKI. The term on-line herein refers to the day-to-day usage of public key certificates and key pairs for authentication. The term off-line herein refers to the more infrequent establishment or dissolution of public key bindings, which may result in the issuing or revocation of public key certificates. For example, the traditional certificate authority is off-line, issues CRLs off-line, and places the CRLs in a directory for on-line retrieval. The scheme involving certificate status proofs also makes use of off-line certificate authorities. The OCSP is the only scheme described above that employs an on-line certificate authority.

For reasons stated above and for other reasons presented in greater detail in the Description of the Preferred Embodiment section of the present specification, there is a need for an improved lightweight PKI that overcomes the above-described revocation problems and is more efficient and more scalable (in terms of numbers of certificates) than conventional PKIs.

Summary of the Invention

The present invention provides a public key infrastructure (PKI) that includes a subject, a verifier, and certificate authority. The certificate authority issues a first unsigned certificate to the subject that binds a public key of the subject to long-term identification information related to the subject. The certificate authority maintains a certificate database of unsigned certificates in which it stores the first unsigned certificate. The verifier maintains a hash table containing cryptographic hashes of valid unsigned certificates corresponding to the unsigned certificates stored in the certificate database and including a cryptographic hash of the first unsigned certificate. The subject presents the issued first unsigned certificate to the verifier for authentication and demonstrates that the subject has knowledge of a private key corresponding to the public key in the unsigned certificate.

In one embodiment, the first unsigned certificate includes an expiration date/time. In one embodiment, the first unsigned certificate does not include an expiration date/time.

In one embodiment, the private key is stored in a secure storage medium accessible by the subject, such as a smartcard or a secure software wallet.

In one embodiment, the verifier computes the cryptographic hash of the first unsigned certificate with a collision-resistant hash function, such as a SHA-1 hash function or a MD5 hash function.

In one embodiment, the certificate authority and the verifier operate to revokes the first unsigned certificate when the binding of the subject's public key to the long-term identification information related to the subject becomes invalid. In one embodiment, the revocation is performed with the following protocol. The certificate authority retrieves first unsigned certificate from the certificate database and computes a cryptographic hash of the first unsigned certificate. The certificate authority sends a message to the verifier containing the cryptographic hash of the first unsigned certificate and requesting that the verifier remove the corresponding cryptographic hash of the first unsigned certificate from its hash table. The verifier removes the cryptographic hash of the first unsigned certificate from its hash table and notifies the certificate authority that it has removed the cryptographic hash of the first unsigned certificate from its hash table. The certificate authority collects the notification sent by the verifier.

In one embodiment, the revocation protocol includes the certificate authority marking the first unsigned certificate in the certificate database as being invalid, for auditing purposes. In an alternate embodiment, the revocation protocol includes the certificate authority deleting the first unsigned certificate from the certificate database.

The PKI according to the present invention is referred to as being a lightweight PKI because it is substantially simpler and more efficient than a conventional PKI that employs traditional long-term certificates. In contrast to the traditional long-term certificates employed by conventional PKIs, the

unsigned certificates for authentication of the subject to the verifier according to the present invention are not signed, do not need an expiration date, and do not use CRLs. The PKI according to the present invention is also more scalable (in terms of numbers of certificates) than a conventional PKI.

Brief Description of the Drawings

Figure 1 is a block diagram of a light-weight public key infrastructure (PKI) according to the present invention employing unsigned certificates.

Figure 2 is a diagram of an unsigned certificate issued from a certificate authority of the PKI of Figure 1.

Figure 3 is a flow diagram of a protocol for issuing an unsigned certificate from a certificate authority of the PKI of Figure 1.

Figure 4 is a flow diagram illustrating a protocol for a subject to demonstrate its identity to a verifier of the PKI of Figure 1.

Figure 5 is a flow diagram illustrating a protocol for a certificate authority to revoke an unsigned certificate for the PKI of Figure 1.

Figure 6 is a block diagram of a computer system and a corresponding computer readable medium incorporating one or more main software program components of a PKI according to the present invention.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present invention. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

A light-weight public key infrastructure (PKI) according to the present invention is illustrated generally at 30 in Figure 1. PKI 30 includes several main components which are each a software program. The main software program

components of PKI 30 run on one or more computer systems. In one embodiment, each of the main software program components runs on its own computer system.

A certificate authority 32 issues unsigned certificates to one or more subjects, such as subject 34. Subject 34 is a user which owns a public key contained in the unsigned certificate and uses the unsigned certificate to demonstrate the subject's identity to one or more verifiers, such as verifier 36, by demonstrating that the subject has knowledge of a private key corresponding to the public key in the unsigned certificate. Verifier 36 is a user which relies on the unsigned certificate presented by subject 34 to verify that subject 34 is the subject of the unsigned certificate and that the attributes contained in the unsigned certificate apply to subject 34.

In some embodiments of PKI 30, the same user can play the role of subject 34 and verifier 36 in different situations. The present invention, however, is particularly beneficial in situations where subjects and verifiers form two distinct classes of users where the class of verifiers is relatively small (e.g., up to a few hundred verifiers) while the class of subjects can be very large (e.g., up to millions of subjects).

PKI 30 does not issue traditional long-term certificates, such as issued by conventional PKIs. Traditional long-term certificates are signed by a certificate authority and bind a public key of the subject with one or more attributes that uniquely identify the subject. Traditional long-term certificates typically have a validity period defined by an expiration date, which is a substantial period from the issue date, such as one year from the issue date. A conventional certificate authority needs to revoke a traditional long-term certificate if the public key certificate binding becomes invalid before the expiration date. As discussed in the Background of the Invention section of the present specification, a variety of methods have been used to revoke traditional long-term certificates, such as by using a certificate revocation list (CRL) or an on-line certificate status protocol (OCSP).

PKI 30 is referred to as being a lightweight PKI because it is substantially simpler and more efficient than a conventional PKI that employs traditional long-term certificates. In contrast to the traditional long-term certificates employed by conventional PKIs, the unsigned certificates for authentication of subject 34 to verifier 36 according to the present invention are not signed, do not need an expiration date, and do not use CRLs. The PKI 30 is also more scalable (in terms of numbers of certificates) than a conventional PKI.

Certificate authority 32 maintains a certificate database 40 containing unsigned certificates. Each unsigned certificate in certificate database 40 binds a public key of a subject, such as subject 34, to one or more attributes that uniquely identify the subject.

Verifier 36 maintains a hash table 42 containing cryptographic hashes of valid unsigned certificates. Each cryptographic hash in hash table 42 is computed from an unsigned certificate using an agreed upon collision-resistant hash function, such as SHA-1 or MD5. Hash table 42 is essentially a list of the currently valid unsigned certificates which is keyed by the cryptographic hash. Cryptographic hashes function well as keys for hash table 42, because cryptographic hashes behave statistically as random quantities.

Subject 34 has a private-key 46 stored in a secure storage medium 48, such as a smartcard or secure software wallet. Subject 34 also has a public key mathematically associated with private key 46 as specified by a public key cryptosystem. Subject 34 registers the public key corresponding to private key 46 with certificate authority 32 by sending the public key and one or more attributes that uniquely identify subject 34's identity to certificate authority 32 and demonstrating that the identification attributes apply to subject 34. Examples of such identification attributes include name, social security number, and employee number.

The components of PKI 30 are linked by one or more computer networks. A network connection 50 couples certificate authority 32 and subject 34. A network connection 52 couples subject 34 and verifier 36. A network connection 54 couples certificate authority 32 and verifier 36. Network

connection 54 is preferably a secure network connection, such as a Secure Sockets Layer (SSL) connection, to properly protect communications between certificate authority 32 and verifier 36. In one embodiment, secure network connection 54 provides data integrity protection, including data origin authentication, and anti-replay protection.

One embodiment of an unsigned certificate issued by certificate authority 32 is illustrated generally at 60 in Figure 2. Unsigned certificate 60 includes a meta-data (MD) field 61 containing data about unsigned certificate 60 itself rather than data related to the subject. Examples of data stored in meta-data field 61 include serial number and issuer name. Unsigned certificate 60 includes a subject's public key (PK) 62. Unsigned certificate 60 includes long-term identification information (LTI) field 63 containing attributes uniquely identifying the subject, such as the subject's name, the subject's social security number, or the subject's employee number.

Unsigned certificate 60 optionally includes a long-term expiration (EXP) field 64 which contains a date/time of expiration for unsigned certificate 60. The expiration date/time contained in long-term expiration field 64 is useful for administrative purposes, but is not required for proper functioning of PKI 30. By contrast, in a conventional PKI, the expiration date is required to reduce the size of the CRL as revoked certificates reach their expiration dates.

A protocol for issuing unsigned certificate 60 from certificate authority 32 is illustrated generally at 100 in Figure 3. At step 102, subject 34 sends its public key and one or more attributes that uniquely identify subject 34 to certificate authority 32.

At step 103, subject 34 demonstrates knowledge of the private key 46 associated with the subject 34's public key. Step 103, is performed in a way that depends on the cryptosystem for which the private-public key pair has been generated by subject 34. For example, in a digital signature cryptosystem, subject 34 demonstrates knowledge of the private key 46 by using private key 46 to digitally sign a quantity derived from a random quantity generated by

certificate authority 32. Certificate authority 32 then verifies this digital signature using subject 34's public key.

At step 104, subject 34 demonstrates to certificate authority 32, by out-of-band administrative means, that the identification attributes sent in step 102 apply to subject 34.

At step 106, certificate authority 32 creates unsigned certificate 60 and stores unsigned certificate 60 in certificate database 40. At step 108, certificate authority 32 sends unsigned certificate 60 to subject 34.

At step 110, certificate authority 32 computes a cryptographic hash of unsigned certificate 60 using an agreed upon collision-resistant hash function, such as SHA-1 or MD5. In step 110, certificate authority 32 sends the computed cryptographic hash of unsigned certificate 60 to verifier 36 over network connection 54, which provides data integrity protection, such as with an SSL connection. Secure network connection 54 prevents an attacker from tricking verifier 36 into accepting, and adding to hash table 42, a cryptographic hash of unsigned certificate 60 that has not been issued by certificate authority 32 or is no longer valid.

At step 112, verifier 36 stores the cryptographic hash of unsigned certificate 60 computed in step 110 in hash table 42.

A protocol employed by subject 34 to demonstrate its identity to verifier 36 is illustrated generally at 200 in Figure 4. At step 202, subject 34 sends a message to verifier 36 containing unsigned certificate 60.

At step 204, if a long-term expiration field 64 is present in unsigned certificate 60, verifier 36 verifies that the expiration date/time specified in long-term expiration field 64 has not been reached.

At step 206, verifier 36 computes a cryptographic hash of unsigned certificate 60 by the agreed upon collision-resistant hash function. In step 206, verifier 36 then verifies that the computed cryptographic hash is present in hash table 42. In step 206, verifier 36 performs a look-up of hash table 42 with an efficient and computationally inexpensive operation. Moreover, since unsigned certificate 60 is not signed, certificate authority 32 does not perform a

computationally expensive signature operation to create unsigned certificate 60 and verifier 36 does not perform a computationally expensive signature verification operation as part of the validation of unsigned certificate 60.

At step 208, subject 34 demonstrates knowledge of the private key 46 associated with the public key 62 contained in unsigned certificate 60. Step 208 is performed in a way that depends on the cryptosystem for which the private/public key pair has been generated by subject 34. For example, in a digital signature cryptosystem, subject 34 demonstrates knowledge of the private key 46 by using private key 46 to digitally sign a quantity derived from a random quantity generated by verifier 36. Verifier 36 then verifies this digital signature using the public key 62 in unsigned certificate 60.

A protocol for revoking unsigned certificate 60 is illustrated generally at 300 in Figure 5. Protocol 300 is performed when subject 34's private key 46 is compromised or the identification attributes in long-term identification information field 63 no longer apply to subject 34, because in either case the binding of public key 62 to the identification attributes is invalid.

At step 302, certificate authority 32 retrieves unsigned certificate 60 from certificate database 40 and computes a cryptographic hash of unsigned certificate 60 using the agreed upon collision-resistant hash function. In one embodiment, certificate authority 32 marks unsigned certificate 60 in certificate database 40 as being invalid, for auditing purposes. In an alternative embodiment, where retention of unsigned certificate 60 in certificate database 40 is not required for auditing purposes, certificate authority 32 deletes unsigned certificate 60 from certificate database 40.

At step 304, certificate authority 32 sends a message to each verifier 36 containing the cryptographic hash of unsigned certificate 60 computed in step 302. The message sent in step 304 also requests that each verifier 36 remove the corresponding cryptographic hash of unsigned certificate 60 from its hash table 42.

At step 306, each verifier 36 removes the cryptographic hash of unsigned certificate 60 from its hash table 42 which corresponds to the cryptographic hash

sent in the message from certificate authority 32 in step 304. In step 306, each verifier 36 notifies certificate authority 32 that it has removed the cryptographic hash of unsigned certificate 60 from its hash table 42. The notification performed in step 306 is sent to certificate authority 32 over secure network connection 54 to prevent an attacker from sending a false notification to certificate authority 32.

At step 308, certificate authority 32 collects the notifications sent by each verifier 36 in step 306. In step 308, certificate authority 32 deems that the revocation of unsigned certificate 60 is completed when notifications are received from all of the verifiers 36.

Once protocol 300 is completed, unsigned certificate 60 will not be accepted by any verifier 36 and will therefore become useless.

Protocol 300 for PKI 30 according to the present invention is practical only when there are a small number of verifiers 36 (e.g., up to a few hundred verifiers). If there are a small number of verifiers 36, it is likely that all of verifiers 36 will receive the request from certificate authority 32 in step 304 of protocol 300, remove the cryptographic hash in step 306, and return the acknowledgment (i.e., notification) in step 308. If one or two verifiers 36 fail to return an acknowledgment, an administrator of certificate authority 32 can investigate and manually correct the cause of the failure.

Figure 6 illustrates one embodiment of a computer system 250 and an external computer readable medium 252 which can be employed according to the present invention to implement one or more of the main software program components of a light-weight PKI according to the present invention, such as PKI 30. Embodiments of external computer readable medium 252 include, but are not limited to: a CD-ROM, a floppy disk, and a disk cartridge. Any one of the main software program components of a light-weight PKI according to the present invention can be implemented in a variety of compiled and interpreted computer languages. External computer readable medium 252 stores source code, object code, executable code, shell scripts and/or dynamic link libraries for any one of the main software program components of a light-weight PKI

according to the present invention. An input device 254 reads external computer readable medium 252 and provides this data to computer system 250. Embodiments of input device 254 include but are not limited to: a CD-ROM reader, a floppy disk drive, and a data cartridge reader.

Computer system 250 includes a central processing unit 256 for executing any one of the main software program components of a light-weight PKI according to the present invention. Computer system 250 also includes local disk storage 262 for locally storing any one of the main software program components of a light-weight PKI according to the present invention before, during, and after execution. Any one of the main software program components of a light-weight PKI according to the present invention also utilizes memory 260 within the computer system during execution. Upon execution of any one of the main software program components of a light-weight PKI according to the present invention, output data is produced and directed to an output device 258. Embodiments of output device 258 include, but are not limited to: a computer display device, a printer, and/or a disk storage device.

Although specific embodiments have been illustrated and described herein for purposes of description of the preferred embodiment, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations calculated to achieve the same purposes may be substituted for the specific embodiments shown and described without departing from the scope of the present invention. Those with skill in the chemical, mechanical, electro-mechanical, electrical, and computer arts will readily appreciate that the present invention may be implemented in a very wide variety of embodiments. This application is intended to cover any adaptations or variations of the preferred embodiments discussed herein. Therefore, it is manifestly intended that this invention be limited only by the claims and the equivalents thereof.